Supporting both exploratory design and power of action with a new property sheet

Raphaël Hoarau, Stéphane Conversy

Supporting both Exploratory Design and Power of Action with a New Property Sheet

Raphaël Hoarau
Université de Toulouse
ENAC - IRIT/ICS
7, rue Edouard Belin
Toulouse, 31055 France
raphael.hoarau@enac.fr

Stéphane Conversy
Université de Toulouse
ENAC - IRIT/ICS
7, rue Edouard Belin
Toulouse, 31055 France
stephane.conversy@enac.fr

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
ACM 978-1-4503-0268-5/11/05.

Abstract
Graphical interaction designers often make tradeoffs between supporting exploratory design and power of action. In this paper, we describe a new property sheet and a set of interaction that aims at supporting both. Thanks to a new visualization of properties and values, and modeless, example-based interaction and selection, designers can make an opportunistic use of implicit groups to augment their power of action.

Keywords
Graphical Interaction Design, Exploratory Design.

ACM Classification Keywords
H.5.2. User Interfaces: Interaction Styles.

General Terms
Design, Human Factors.

Introduction
One of the main activities that relies on interactive graphics is *exploratory design* [2], e.g. sketching; designing slides for a presentation, designing class hierarchy, etc. where the final product cannot be envisaged and has to be discovered while being

designed. During exploratory design, users go back and forth, trying and canceling solutions on a partial set of graphical objects. Once users are satisfied with their design, they have to apply the newly designed properties to other existing objects. If the system does not support this task, users are required to repeat actions to propagate a change (problem known as viscosity [2]). Some tools (such as "masters" in presentation software or CAD [5]) enable users to apply modifications on multiple objects. However, such concepts compete with exploratory design: they imply premature commitment [2] when the nature of the structure has to be discovered, or are cumbersome to define and apply after having designed multiple objects (viscosity problem [2]).

Hence, designing requires a system that offers fluidity and rapid manipulation, so as to foster exploratory design. And designing also requires a system that fosters power of action, i.e. the extent to which a user can modify a number of objects in a minimum of actions (possibly at once). This paper introduces a new instrument and new interactions that aim at fostering both exploratory design and power of action. This instrument is a new kind of property sheet, i.e. a window containing a vertical list of pairs of property type and value (e.g. shape: rectangle, color: green, thickness: 3…). Thanks to a new visualization of properties and values, and modeless [3][4], example-based interaction and selection, designers can make an opportunistic use of implicit (i.e. unplanned) groups to augment their power of action.

Context

We have designed a graphical drawing application to illustrate the new property sheet. There are four parts: the tool palette on the left side, the workspace in the middle, the sample panel on top right, and the property sheet on bottom right (see **figure 1**).

The workspace is the main view, where users can create a new object, by clicking and resizing. Selection is performed by clicking on an object or by drawing a rubber rectangle to encompass several items at once, as done with usual graphics editors. Selected items are highlighted with a shadow, while other items are made translucent.

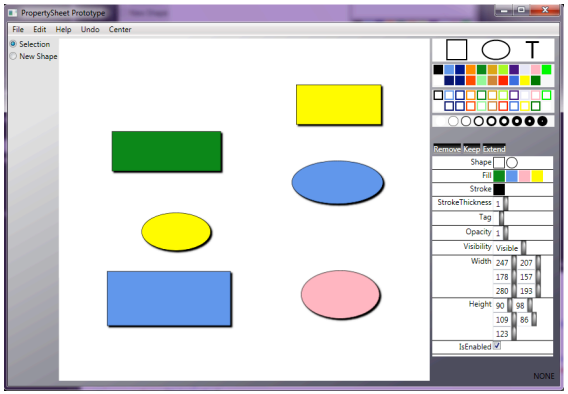

figure 1 Overview of the application. The workspace is at the center, the samples at the top right and the property sheet at the bottom right.

The samples panel contains a set of values for shape, fill color (represented by a colored square), stroke color (repr. by a stroked-only colored square) and stroke thickness (repr. by a stroked-only circle). In order to modify a property of an object in the main view, users can drag a sample and drop it on the object. Feedback is shown as soon as the object is hovered, in order for the user to understand the action, and to assess the

change before effectively applying it by releasing the mouse button. This enables to cancel the action, by releasing the button outside of any object. Drag and drop of samples also apply on a selection with multiple objects. The interactions described so far are not new. Next section presents the property sheet with novel interactions.

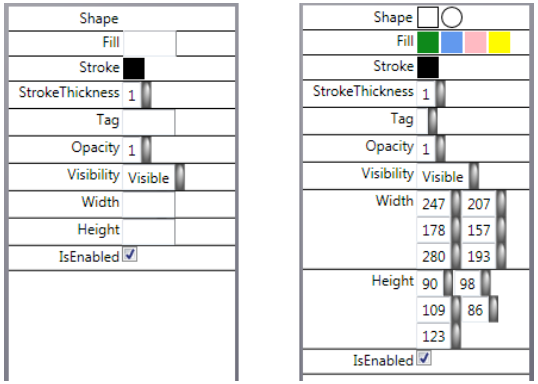

figure 2. The user's selection contains objects with varying shapes, fill colors, width, and height. A classical property sheet (left) displays a blank fill for those properties, whereas our property sheet (at right) displays all different values.

The property sheet

A property sheet offers two services to the user: visualize values (with progressive disclosure [4]), and modify them [4]. In classical property sheets, if multiple objects are selected, only "shared values" (i.e. values shared by all objects) are shown and are modifiable (see **figure 2**, left). Users can change a shared value for a property type, and the system reflects change to all selected objects (power of action). Other properties, those which are multi-valuated, still appear, but have their value set to blank, and cannot be modified. Those blank fills limit visualization (they don't inform users with the values) and power of action (they don't allow users modifying them precisely for multiple objects).

Our version of the property sheet differs in that it shows all values for a multi-valuated property (see **figure 2**, right), instead of displaying nothing. We relied on the display of those values to design a set of interactions that offer new services for exploratory design and power of action: query and selection of objects with graphical example, refine selection, modify properties on multiple objects with precision, and structure the scene.

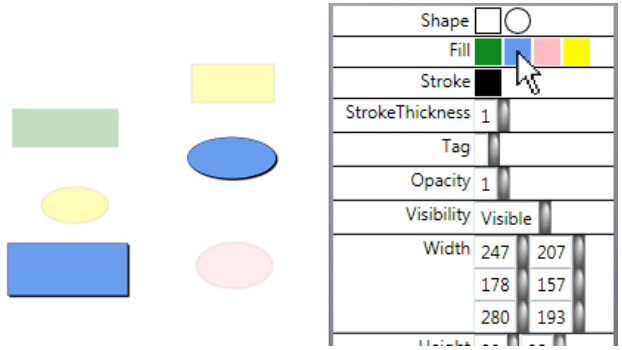

figure 3. The user's cursor is over the blue shared value of the fill property (fill: blue). Because they don't have this shared value, the green rectangle, the pink circle and the two yellow shapes are dimmed.

The representation of a shared value in the property sheet actually refers to two concepts: the value in itself, and the set of selected objects that exhibits this property value. As a value per se, and similarly to the interaction with the samples panel, users can drag the representation (considered as a value) from the property sheet onto (a selection of) objects in the main

view to modify a property. If the shared value is numerical, users can hover over it and rotate the mouse wheel to increment or decrement it (*power* and *precision*). Together with immediate feedback, this enables both exploration and precise adjustment of properties, thus reducing temporal offset [1] between action and reaction.

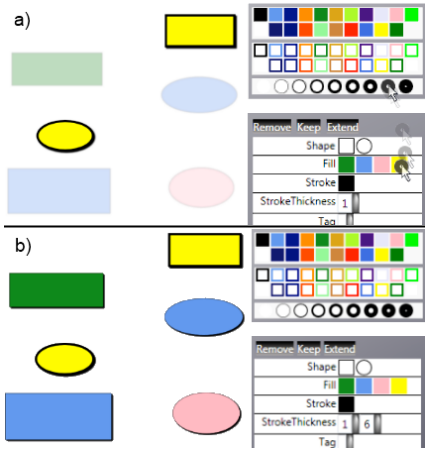

figure 4. a) The user drags the "thickness: 6pt stroke" sample over the "fill: yellow" shared value. a) Immediate feedback turns all yellow items' stroke thickness to 6pt. b) The user has dropped the sample, the modification is applied.

Since the representation of a shared value also reifies [1] a set of objects, hovering over a shared value makes the concerned objects highlighted while others are blurred (**figure 3**). This makes it easy to figure out what set are made of what, and possibly detects outliers (*structuring*). In addition, users can drag a sample (hence a value) from the sample panels onto the representation of a shared value (considered as a set of objects) in the property sheet to modify at once a property onto multiple objects (*power*) (**figure 4**).

Users can also drag a representation from (value) and in (set) the property sheet (**figure 5**). Immediate feedback during interaction helps user assessing the result of their actions, while possibly cancelling it, as explained above. These interactions can be considered as queries that help define sets of objects with implicit group based on graphical properties (*power of action*).

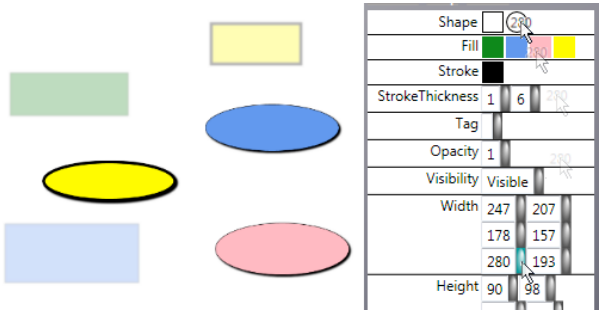

figure 5. The user drags the "width: 280" shared value and drops it on the "shape: circle" shared value. All circles in the selection now have a width set to 280.

To select objects, users can directly click on them in the workspace, or draw a selection rectangle. In order to refine the selection, users can utilize three meta-instruments (i.e. instrument that control instrument, here the selection): *Remover, Keeper and Extender*. The interaction consists in a drag and drop of the representation of the instrument on a shared value. *Remover* throws out of the selection all objects that have this shared value (**figure 6**). *Keeper* keeps in the selection the objects that have this shared value, and throws away the others. *Extender* adds to the selection all objects that are not selected but possess this shared value. These interactions extend the set of example-based queries introduced above (*power of action*).

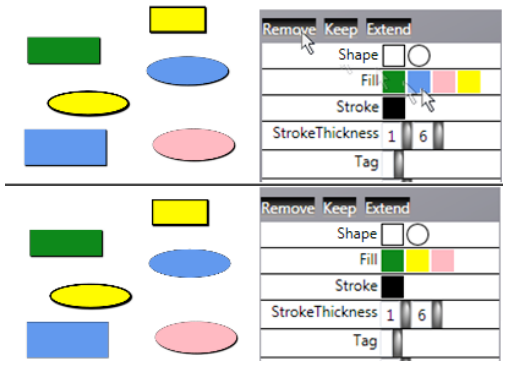

figure 6. The user drags the Remove instrument and drops it on the blue fill shared value. All blue objects are removed from the selection.

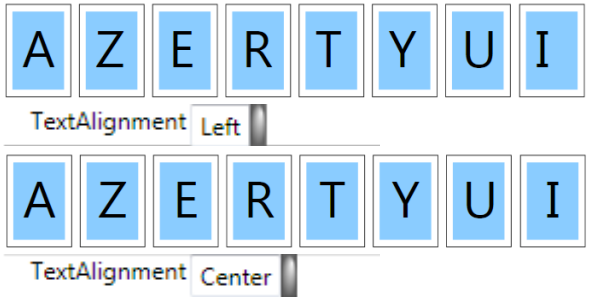

Figure 7. The text alignment is set to left for each letter instead of centered. The difference is not really perceptible for larger letters but noticeable on the 'I' key.

Scenario

This section describes a scenario that illustrates the usefulness of our property sheet. The user of a graphic design tool has to create a virtual keyboard. After having realized the first key ('A'), she duplicates it and changes the letter on the others. When editing the 'I' key, she realizes that the text is not centered on the key (*exploratory design*). She has to change it for all the duplicated keys: she uses the "Extend" instrument on the "text-alignment: left value" to add all others letters which are left-aligned to the selection, and sets the "alignment" to "centered" (**Figure 7**). (*selection refinement, power, use of implicit group*).

After having designed the entire keyboard, she found that the text of keys with two characters is too small (*exploratory design*). She selects one of them, and uses the wheel on the font-size text box to increase the size (**figure 9**) until she gets a satisfying result (*exploration* and *precise adjustment*).

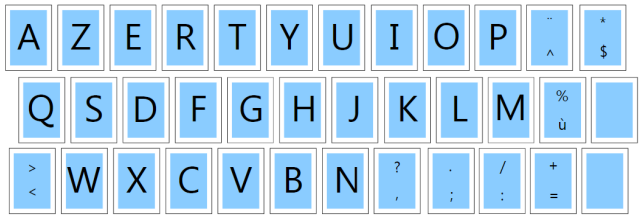

figure 8. The entire keyboard.

figure 9. The user has selected the '?' text by clicking on it, then he increases the font-size by using the mouse wheel on the font-size textbox in the property sheet.

The user draws a selection rectangle on the keyboard (*visualization of values*), and she verifies that there are three different shared values for the font-size property (*structuring*). The user still has to set the new font-size

to all other "double characters" keys. She drags the larger font-size and drops it on the smaller one, which turns smaller text larger (*power*).

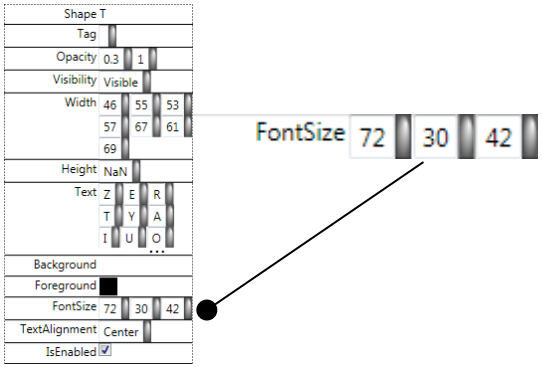

figure 10. The user has selected the entire keyboard. There are three font-size shared by the selected objects: 72 is for each text in the single character keys, 30 for each text in the double character keys and 42 the new font-size of the '?' text.

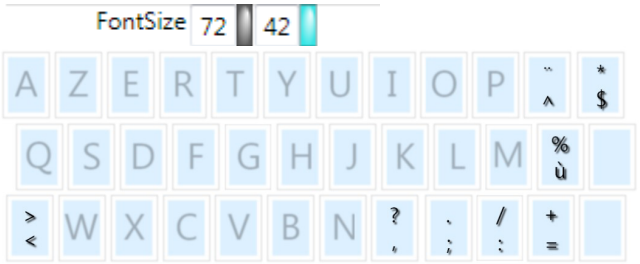

figure 11. The user has dragged the 42 font-size value on the 30. All the elements which had the 30 font-size now have the 42 one.

In summary, the user has created the keyboard with no tasks planning, only by exploration. There was no need to create groups (or "masters") prematurely, yet she was still able to make powerful actions by changing some parts of the duplicated elements at once. To do so, she relies on modeless direct manipulation [4] and an opportunistic use of implicit groups reified into visible, graphical properties. The immediate feedback of her actions helped to understand what was going to change, and to compare two different states of her work.

Conclusion

We have presented a new kind of property sheet to enable both powerful actions and exploratory design. We are currently improving it with other interaction techniques, and plan to conduct evaluation on their effectiveness as a support for exploratory design.

Acknowledgements

We thank all members of the LII team involved in the participatory design sessions that led to this design.

References

[1] Beaudouin-Lafon, M. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. In Proc. CHI'00. (2000), 446-453.

[2] Green T.R.G, Cognitive dimensions of notations, in People and Computers V, A Sutcliffe and L Macaulay, Eds. Cambridge : CUP, 1989 pp. 443-460.

[3] Johnson J.A., Roberts T.L., Verplank W., Smith D.C., Irby C.H., Beard M., and Mackey K. The Xerox Star: A retrospective. IEEE Computer, 22(9):11–29, 1989.

[4] Shneiderman, B. (1983). Direct manipulation: a step beyond programming languages. IEEE Computer 16, 8, 57–69.

[5] Sutherland, I.E. Sketchpad, A Man-Machine Graphical Communication System, 1963, PhD thesis, Massachusetts Institute of Technology.